HN

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/082,807	02/22/2002	David Bau III	109870-130103	2054

25943 7590 01/14/2005

SCHWABE, WILLIAMSON & WYATT, P.C.
PACWEST CENTER, SUITES 1600-1900
1211 SW FIFTH AVENUE
PORTLAND, OR 97204

EXAMINER	
HOANG, PHUONG N	

ART UNIT	PAPER NUMBER
2126	

DATE MAILED: 01/14/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/082,807	BAU ET AL.	
	Examiner	Art Unit	
	Phuong N. Hoang	2126	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *22 February 2002*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1 - 84* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1 - 84* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date *11/21/02 & 4/9/03*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1 – 84 are pending for examination.

Claim Rejections - 35 USC § 101

2. Claims 12 – 19, and 30 – 42 are rejected under 35 U.S.C. 101 because they are directed to non-statutory subject matter.

3. Claims 12 – 19, and 30 – 42 are directed to method steps which can be practiced mentally in conjunction with pen and paper, therefore they are directed to non-statutory subject matter. Specifically, as claimed, it is uncertain what performs each of the claimed method steps. Moreover, each of the claimed steps, inter alia, receiving, parsing, storing, and invoking, can be practiced mentally in conjunctions with pen and paper. The claimed steps do not define a machine or computer implemented process [see MPEP 2106]. Therefore, the claimed invention is directed to non-statutory subject matter.

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

5. Claims 1 – 2, 10, 30 – 32, 35 – 36, 38 – 39, 41 – 44, 52, 72 – 74, 77 – 78, 80 – 81, and 83 – 84 are rejected under 35 U.S.C. 103(a) as being unpatentable over Schofield, US patent no. 6,253,252.

6. Schofield was cited by applicant in IDS filed on 11/21/02.

7. **As to claim 1**, Schofielf teaches a method of specifying an asynchronous web service within a procedural programming environment, the method comprising the steps of:

providing a source code (code generator, col. 7) representation of at least a portion of web service (Internet, col. 4 lines 62 – 66) logic, the logic including at least one method declared to be a callback method (callback or response method, col. 4 lines 12 – 20, col. 8 lines 14 – 23, and col. 11 lines 20 - 52), and

identifying a array (array containing address, col. 10 – lines 1 – 10 and col. 12) declared to implement said callback method to cause a compiler and code generator (compiler produce compiled client stub, col. 7 lines 12 – 60) to generate a client proxy object (proxy or stub, col. 7 lines 12 – 40) for interacting asynchronously (asynchronously calling, abstract and col. 7 lines 50 – col. 8) with the client using said callback method, and to assign the client proxy object to said member variable (assign proxy an address, col. 9 lines 32 – col. 10 lines 20).

Schofield does not explicitly teach a member variable.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to recognize that the array is also a variable declared to implement the response.

8. **As to claim 2**, Schofield teaches the step of wherein said callback method is declared inside a callback definition (the callback is defined, col. 7 lines 50 – 67).

9. **As to claim 10**, Schofield teaches a procedural programming environment, a method of returning an asynchronous response (return asynchronous calling, abstract and col. 7 lines 50 – col. 8) to a client, the method comprising:

identifying a callback (callback or response, col. 4 lines 12 – 20, col. 8 lines 14 – 23, and col. 11 lines 20 - 52) associated with the client,

identifying an array (array variable containing address, col. 10 – lines 1 – 10 and col. 12) declared to implement the callback;

generating a proxy object (proxy or stub, col. 7 lines 12 – 40) that implements the callback interface (stub functions .... Interface, col. 8 lines 5 – 15); and

assigning the proxy object to the member variable (assign proxy an address, col. 9 lines 32 – col. 10 lines 20).

Schofield does not explicitly teach member variable.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to recognize that the array is also a variable declared to implement the response.

10. **As to claim 30**, Schofield teaches a web service, a method comprising the steps of:

generating a request (call from client through the internet, col. 10 lines 50 – 65 and col. 4 lines 65 – 67) to an external web service using a proxy object (compiler produced client stub object, col. 7 lines 25 – 40), previously generated by a compiler based upon a service description file (source file 101, col. 7 lines 10 – 60), wherein the request includes a callback address (containing address, col. 10 – lines 1 – 10 and col. 12) to identify a location to which the external web service should return a response,

transmitting the request as a request message to the external web service using one or more transmission protocols (network protocol, col. 6 lines 14 – 17) and

receiving an asynchronous response from the external web service (a response is returned to client, col. 11 lines 20 – 30 and col. 8 lines 18 – 22).

Schofield does not explicitly teach that the description file associated with the external web service. However, Schofield teaches the description file is compiled and linked to client and server (col. 7 lines 10 – 60).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to recognize that the file has to be associated with the external web service to link the client application to produce the complied code to be running.

11. **As to claim 31**, Schofield teaches the step of wherein the callback address includes proxy object identifier (store address in the proxy, col. 9 lines 35 – 40).

12. **As to claim 32**, Schofield teaches the step of wherein the callback address is included within one or more headers of the request message (header file 119, col. 7. lines 40 – 50).

13. **As to claim 35**, Schofield teaches the step of a callback instance identifier representing a specific instance of the requesting web service to which asynchronous responses are to be routed (asynchronous response, col. 8 lines 45 – 65).

14. **As to claim 36**, Schofield teaches the step of wherein and the callback instance identifier is included within one or more headers of the request message (header file 119, col. 7. lines 40 – 50).

15. **As to claim 38**, Schofield teaches a web service, a method comprising the steps of:

receiving a message (receive the message transmitted from client-side, col. 10 lines 10 – 15) identifying a callback address (the object address call's input parameters, col. 9 lines 55 – col. 10 lines 35) including a callback method (object's method to be called), and a proxy object identifier (proxy handle);

extracting the proxy object identifier from the message,

determining a method to be invoked based at least in part upon the proxy object

identifier (determined by the proxy handle) and the callback method (used input parameters and carried out the request), and

calling the method to be invoked (call the appropriate method).

Schofield teaches calling the method. Schofield does not explicitly teach routing to the method.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to recognize that calling the method would comprise routing to the method to be called.

16. **As to claim 39**, Schofield does not teach the step of wherein the callback address comprises a URL indicating a location where the web service listens for callbacks from the external web service.

It would have been obvious to one o of ordinary skill in the art at the time the invention was made to recognize that the web address would comprise the URL.

17. **As to claim 41,** Schofield teaches the step of wherein the message further identifies a callback instance identifier (call identifier, col. 8 lines 35 – 45 and col. 13 lines 25 – 35).

18. **As to claim 42,** Schofield teaches the step of wherein routing further comprises identifying a callback instance based at least in part upon the callback instance identifier

(specified object to be called, col. 9 lines 55 – col. 10 lines 10); and routing the request to a method associated with the identified callback instance.

19. **As to claim 43**, this is the article claim of claim 1. See rejection for claim 1 above.

20. **As to claim 44,** see rejection for claim 2 above.

21. **As to claim 52**, this is the article claim of claim 10. See rejection for claim 10 above.

22. **As to claim 72**, this is the article claim of claim 30. See rejection for claim 30 above.

23. **As to claims 73 - 74,** see rejection for claims 32 - 33 above.

24. **As to claims 77 - 78,** see rejection for claims 35 - 36 above.

25. **As to claim 80**, this is the article claim of claim 38. See rejection for claim 38 above.

26. **As to claim 81,** see rejection for claim 39 above.

27. **As to claims 83 - 84,** see rejection for claims 41 – 42 above.

28. Claims 3 – 8, 11, 45 – 50, and 53 are rejected under 35 U.S.C. 103(a) as being unpatentable over Schofield, US patent no. 6,253,252 in view of Chan, US patent no. 6,230,160.

29. Chan was cited by applicant in IDS filed on 11/21/02.

30. **As to claim 3**, Schofield teaches the step of specifying one or more declarative annotations (parameters, col. 7 lines 50 – 65) associated with said callback method.

Schofield does not explicitly teach the step of a compiler to generate one or more persistent components to maintain conversational state related to the identified member variable.

Chan teaches the step of components maintaining state information (state information .....event proxy object, col. 7 lines 25 – 40 and col. 9 lines 20 – 30).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teaching of Schofield and Chan's system because Chan's maintaining state information would provide information to retrieve all information received from either the client or server.

31. **As to claim 4**, Schofield teaches the step of wherein the one or more declarative annotations (the method uses the input parameters and carried out the request, col. 10 lines 20 – 50) indicate to the compiler whether the identified method is at least one of a start method (start, col. 6 lines 18 - 60), a continue method (continue, col. 11 lines 35 – 45), and a finish (complete, col. 10 lines 38 – 50) method, wherein the start method applies to the start of a stateful conversation between the client and the web service, the continue method applies to the continuation of an ongoing stateful conversation between the client and the web service, and the finish method applies to the completion of an ongoing stateful conversation between the client and the web service.

32. **As to claims 5 - 7**, Schofield teaches the step of wherein when a method declared to be a start method is invoked at run-time (system starts, col. 6 lines 18 – 67), a new instance of a conversation is created (calls), and a unique identifier (unique identifier) is associated with that conversational instance to facilitate management of multiple simultaneous conversations (conversation between different routines).

33. **As to claim 8**, wherein when a method declared to be a finish method is invoked at run-time, the corresponding instance of the conversation is destroyed (destroyed, col. col. 6 lines 20 – 30) after processing by the web service logic.

34. **As to claim 11**, see rejection for claim 3 above.

35. **As to claims 45 – 50**, see rejection for claims 3 - 8 above.

36. **As to claim 53**, see rejection for claim 3 above.

37. Claims 9 and 51 are rejected under 35 U.S.C. 103(a) as being unpatentable over Schofield, US patent no. 6,253,252 in view of Chan, US patent no. 6,230,160, and further in view of Ben-Shachar, US patent no. 6,209,018.

38. Ben-Shachar was cited by applicant in IDS filed on 11/21/02.

39. **As to claim 9**, Schofield and Chan do not explicitly teach one or more queues to temporarily store one or more asynchronous responses for delivery to the client when the client is able to receive the responses.

Ben-Shachar teaches the step of one or more queues (queues, fig. 4 and 26 and col. 11 lines 50 - 60) to temporarily store one or more asynchronous responses for delivery to the client when the client is able to receive the responses.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teaching of Schofield, Chan, and Ben-Shachar's system because Ben-Shachar's first-in and first out queues are well known to be useful and efficient to hold multiple asynchronous requests or responses for delivering to the client.

40. **As to claim 51**, see rejection for claim 9 above.

41. Claims 12 – 23, 33 – 34, 37, 40, 54 – 65, 75 – 76, 79, and 82 are rejected under 35 U.S.C. 103(a) as being unpatentable over Schofield, US patent no. 6,253,252 in view of Grant, US Pub no. 20020099738.

42. Grant was cited by applicant in IDS filed on 04/08/03.

43. **As to claim 12**, Schofield teaches an asynchronous web service, a method comprising the steps of:

receiving a message from a client (receiving the request transmitted from client, col. 10 lines 10 – 20) requesting that a web service method be invoked;

a callback address (the address for the response, col. 8 lines 23 – 35, col. 10 lines 1 – 10) indicating a location where the client is listening for callbacks from the web service;

storing the callback address in association with a previously generated proxy object (store the routine address in the proxy, col. 9 lines 35 - 40);

invoking the requested web service method (invokes, col. 3 lines 1 – 18).

Schofield does not explicitly teach the steps of parsing the message, storing the callback address in association with a previously generated proxy object.

Grant teaches the step of parsing the message (parse the message, col. 3 [0037] and [0041].

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teaching of Schofield and Grant's system because Grant's parsing the input message is well known to be necessary for the server to understand the content of the message.

44. **As to claim 13**, Schofield teaches the steps comprising of:

identifying an instance identifier (identifier, col. 4 lines 1 – 10, and col. 13 lines 11 – 40) provided by the client indicating a particular instance of the client that is listening for callbacks from the web service; and

storing (store the output parameters which comprises callback address, col. 13 lines 12 – 60 and col. 9 lines 55 – col. 10 lines 20) the instance identifier in association with the callback address and the proxy object.

45. **As to claims 14 and 15**, Schofield teaches the step comprising of wherein at least one of the callback address and the instance identifier is encapsulated in one or more request messages (each request has an identifier, col. col. 4 lines 1 – 10, and col. 13 lines 11 – 40).

46. **As to claim 16,** Schofield modified by Grant teaches the step of wherein at least one of the callback address and the instance identifier is encapsulated in one or more SOAP message headers (Grant; SOAP message headers, [0037] and [0276]).

47. **As to claim 17**, Schofield teaches the step of wherein the instance identifier is a GUID (identifier is unique, col. col. 4 lines 1 – 10, and col. 13 lines 11 – 40).

48. **As to claim 18**, Schofield modified by Grant teaches wherein invoking the requested web service method further comprises the steps of

extracting data representation language elements from the message (Grant; soap message is parse by xml engine, [0037] – [0041] ;

mapping the data representation language elements to programming language objects (Grant; mapping XML in SOAP message into an appropriate CORBA invocation, [0038], [0040] – [0042], [0047], and [0049]);

invoking the requested web service method including passing the programming language objects as parameters to the web service method (Grant; send the corba invocation to corba object, [0040] - [0041]);

mapping programming language objects returned by the web service (Grant; corba object sends response back to xml engine, [0042]) onto representative data representation language elements (Grant; convert the result into xml, [0042]);

packaging the representative data representation language elements according

to one or more protocols (Schofield; known networking protocols, col. 6 lines 10 – 17) used by the client in generating the requested message; and

transmitting the packaged data representation elements to the client (Grant; send the resulting to web client, [0042]) in accordance with one or more protocols used by the client in transmitting the message to the web service.

49. **As to claim 19**, Schofield modified by Grant teaches the steps comprising of

generating an asynchronous response to the client in response to the message (Schofield; return asynchronous calling, abstract and col. 7 lines 50 – col. 8) by invoking a method on the previously generated proxy object using a declared member variable, wherein invoking includes

passing programming language objects as parameters (Schofield; parameters, col. 9 lines 55 – col. 10 lines 10) to said method,

mapping the programming language objects and method invocation onto representative data representation language elements (Grant; translates the request into XML, [0041]),

packaging the representative data representation language elements (Grant; XML, [0042]), and callback instance identifier (Schofield; identifier, col. 4 lines 1 – 10, and col. 13 lines 11 – 40) into an asynchronous response message, and

transmitting the asynchronous response message to the client at the callback address in accordance with one or more protocols (Schofield; protocols, col. 6 lines 10 – 16) used by the client in transmitting the original request message to the web service.

50. **As to claim 20**, Schofield teaches a method for specifying logic within a procedural programming environment for receiving a callback from an asynchronous web service, the method comprising the steps of:

identifying a array (array containing address, col. 10 – lines 1 – 10 and col. 12) to be used for interacting with said asynchronous web service (aynchronous call, abstract, and col. 9 - 10) ; and

providing a method (object's method, col. 9 lines 30 – col. 10 lines 10) associated with said member variable, the method containing logic for receiving said callback (parameters containing address to return requests) from said asynchronous web.

Schofield does not explicitly teach the member variable, and method having a signature.

Grant teaches the method having a signature (signature, [0016]).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teaching of Schofield and Grant's system because Grant's signature would contain the description name of the address which is used to return the response, and the array is also a variable declared to implement the response.

51. **As to claim 21**, Scholfield teaches the step of wherein the method is manually provided by a developer (it is the logic of the program).

52. **As to claim 22**, Grant teaches the step of wherein the method signature is provided by an integrated development environment based on a specified service description file (the description of the address, [0016]) containing a declaration for said callback.

53. **As to claim 23**, Grant teaches wherein said method is associated with said member variable using a method naming convention (w3c is the name of the company, [0016]) that utilizes the name of said member variable and the name of said callback.

54. **As to claim 33**, Grant teaches the step of wherein the request message is a SOAP based message (soap message, fig. 3).

55. **As to claim 34,** Grant teaches wherein the callback address comprises a URL (url, [0267]) identifying a location where the web service is listening for a response from the external web service.

56. **As to claim 37**, see rejection for claim 33 above.

57. **As to claim 40**, Schofield modified by Grant teaches the step of wherein determining the method to be invoked comprises appending a name (Grant;

namespace, [0010] – [0119] associated with the callback method to the proxy object identifier.

58. **As to claim 54**, this is the article claim of claim 12. See rejection for claim 12 above.

59. **As to claims 55 – 61**, see rejection for claims 13 – 19 above.

60. **As to claim 62**, this is the article claim of claim 20. See rejection for claim 20 above.

61. **As to claims 63 - 65**, see rejection for claims 21 – 23 above.

62. **As to claims 75 - 76**, see rejection for claims 33 – 34 above.

63. **As to claim 79**, see rejection for claim 37 above.

64. **As to claim 82**, see rejection for claims 40 above.

65. Claims 24 – 29, and 66 – 71 are reject d under 35 U.S.C. 103(a) as being unpatentable over Schofield, US patent no. 6,253,252 in view of Grant, US Pub no. 20020099738, and further in view of Chan, US patent no. 6,230,160.

66. **As to claim 24**, Schofield teaches the step of specifying one or more declarative annotations (parameters, col. 7 lines 50 – 65) associated with said member variable.

Schofield does not explicitly teach the step of a compiler to generate one or more persistent components to maintain conversational state related to the asynchronous web service.

Chan teaches the step of components maintaining state information (state information .....event proxy object, col. 7 lines 25 – 40 and col. 9 lines 20 – 30).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teaching of Schofield and Chan's system because Chan's maintaining state information would provide information to retrieve all information received from either the client or server.

67. **As to claim 25**, Schofield teaches the step of wherein the one or more declarative annotations are specified within the source code (they are within the source code, col. 7 lines 50 – 67).

68. **As to claim 26**, Grant teaches the step of wherein the one or more declarative annotations are specified outside of the source code (attached, [0014] – [0015]).

69. **As to claim 27**, Schofield modified by Grant teaches the step of wherein the one or more declarative annotations are manually specified by a developer (Grant; each parameter is annotated, [0014]).

70. **As to claim 28**, Schofield teaches the step of wherein the one or more declarative annotations are automatically (code generator, col. 7) specified by an integrated development environment based upon input provided by a developer.

71. **As to claim 29,** Schofield teaches the step of wherein asynchronous responses from the external web service are passed to said method associated with said member variable (up-calls the appropriate method based upon the address, col. 8 lines 15 – 25).

72. **As to claims 66 - 71**, see rejection for claims 24 - 29 above.

Conclusion

73. The prior art made of record but not relied upon request is considered to be persistent to applicant's disclosure.

Ben-Sharchar et al., US patent no. 6,209,018, demonstrating a framework for a distributed object network system.

74. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Phuong N. Hoang whose telephone number is (571)272-3763. The examiner can normally be reached on Monday - Friday 9:00 am to 5:30 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai An can be reached on (571)272-3756. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Ph
January 7, 2005

MENG-AI T. AN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100